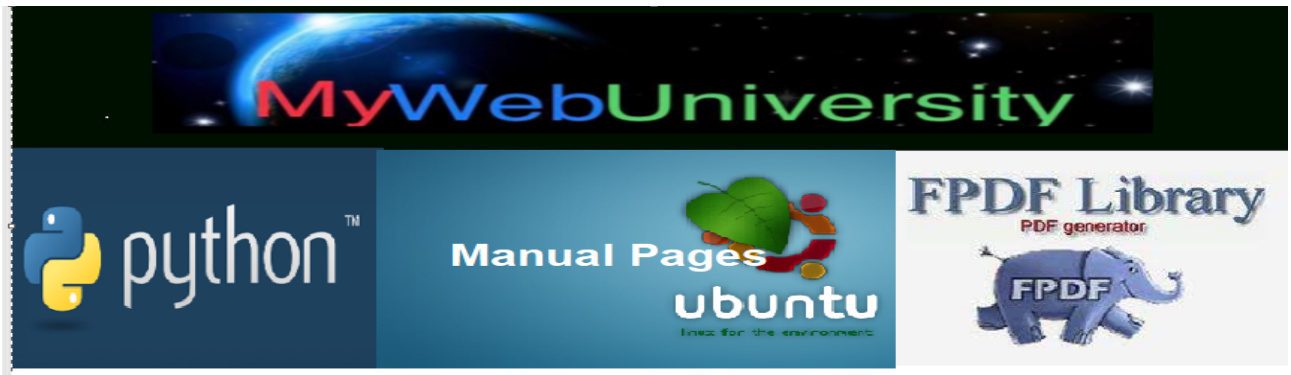

Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 26.04 Manual Pages on command 'isympy.1'

$ man isympy.1

isympy(1) isympy(1)

NAME

isympy - interactive shell for SymPy

SYNOPSIS

isympy [-c | --console] [-p ENCODING | --pretty ENCODING] [-t TYPE | --types TYPE] [-o ORDER | --order ORDER] [-q | --quiet] [-d | --doctest] [-C | --no-cache] [-a | --auto] [-D | --debug] [-- | PYTHONOPTIONS]

isympy [{-h | --help} | {-v | --version}]

DESCRIPTION

isympy is a Python shell for SymPy. It is just a normal python shell (ipython shell if you have the ipython package installed) that executes the following commands so that you don't have to:

>>> from __future__ import division

>>> from sympy import *

>>> x, y, z = symbols("x,y,z")

>>> k, m, n = symbols("k,m,n", integer=True)

So starting isympy is equivalent to starting python (or ipython) and executing the above commands by hand. It is intended for easy and quick experimentation with SymPy. For more complicated programs, it is recommended to write a script and import things explicitly (us? ing the "from sympy import sin, log, Symbol, ..." idiom).

OPTIONS

-c SHELL, --console=SHELL

Use the specified shell (python or ipython) as console backend instead of the default

one (ipython if present or python otherwise).

Example: isympy -c python

SHELL could be either 'ipython' or 'python'

-p ENCODING, --pretty=ENCODING

Setup pretty printing in SymPy. By default, the most pretty, unicode printing is en?

abled (if the terminal supports it). You can use less pretty ASCII printing instead

or no pretty printing at all.

Example: isympy -p no

ENCODING must be one of 'unicode', 'ascii' or 'no'.

-t TYPE, --types=TYPE

Setup the ground types for the polys. By default, gmpy ground types are used if gmpy2

or gmpy is installed, otherwise it falls back to python ground types, which are a

little bit slower. You can manually choose python ground types even if gmpy is in?

stalled (e.g., for testing purposes).

Note that sympy ground types are not supported, and should be used only for experi?

mental purposes.

Note that the gmpy1 ground type is primarily intended for testing; it the use of gmpy

even if gmpy2 is available.

This is the same as setting the environment variable SYMPY_GROUND_TYPES to the given

ground type (e.g., SYMPY_GROUND_TYPES='gmpy')

The ground types can be determined interactively from the variable sympy.polys.do?

mains.GROUND_TYPES inside the isympy shell itself.

Example: isympy -t python

TYPE must be one of 'gmpy', 'gmpy1' or 'python'.

-o ORDER, --order=ORDER

Setup the ordering of terms for printing. The default is lex, which orders terms lex?

icographically (e.g., x**2 + x + 1). You can choose other orderings, such as rev-lex,

which will use reverse lexicographic ordering (e.g., 1 + x + x**2).

Note that for very large expressions, ORDER='none' may speed up printing consider?

ably, with the tradeoff that the order of the terms in the printed expression will

have no canonical order

Example: isympy -o rev-lax

ORDER must be one of 'lex', 'rev-lex', 'grlex', 'rev-grlex', 'grevlex', 'rev-

grevlex', 'old', or 'none'.

-q, --quiet

Print only Python's and SymPy's versions to stdout at startup, and nothing else.

-d, --doctest

Use the same format that should be used for doctests. This is equivalent to 'isympy -c python -p no'.

-C, --no-cache

Disable the caching mechanism. Disabling the cache may slow certain operations down considerably. This is useful for testing the cache, or for benchmarking, as the cache can result in deceptive benchmark timings.

This is the same as setting the environment variable SYMPY_USE_CACHE to 'no'.

-a, --auto

Automatically create missing symbols. Normally, typing a name of a Symbol that has not been instantiated first would raise NameError, but with this option enabled, any undefined name will be automatically created as a Symbol. This only works in IPython 0.11.

Note that this is intended only for interactive, calculator style usage. In a script that uses SymPy, Symbols should be instantiated at the top, so that it's clear what they are.

This will not override any names that are already defined, which includes the single character letters represented by the mnemonic QCOSINE (see the "Gotchas and Pitfalls" document in the documentation). You can delete existing names by executing "del name" in the shell itself. You can see if a name is defined by typing "'name' in glob? als()".

The Symbols that are created using this have default assumptions. If you want to place assumptions on symbols, you should create them using symbols() or var().

Finally, this only works in the top level namespace. So, for example, if you define a function in isympy with an undefined Symbol, it will not work.

-D, --debug

Enable debugging output. This is the same as setting the environment variable SYMPY_DEBUG to 'True'. The debug status is set in the variable SYMPY_DEBUG within isympy.

These options will be passed on to ipython (1) shell. Only supported when ipython is being used (standard python shell not supported).

Two dashes (--) are required to separate PYTHONOPTIONS from the other isympy options.

For example, to run iSymPy without startup banner and colors:

isympy -q -c ipython -- --colors=NoColor

-h, --help

Print help output and exit.

-v, --version

Print isympy version information and exit.

FILES

${HOME}/.sympy-history

Saves the history of commands when using the python shell as backend.

BUGS

The upstreams BTS can be found at ?https://github.com/sympy/sympy/issues? Please report all bugs that you find in there, this will help improve the overall quality of SymPy.

SEE ALSO

ipython(1), python(1)

2007-10-8 isympy(1)